Listing Networks

```
PS C:\Users\JervinJosh\Documents\Github\ME8_Containerization_and_Docker\5-volumes> docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
f426e71b7340   bridge    bridge    local
54d52346937f   host      host      local
0a0f7386858c   none      null      local
```

The default bridge network

1. Inspecting the bridge network

```
PS C:\Users\JervinJosh\Documents\Github\ME8_Containerization_and_Docker\5-volumes> docker network inspect bridge
[
    {
        "Name": "bridge",
        "Id": "f426e71b734051434c11ddc409a80aec425dd57c8598c856f193290953e6362f",
        "Created": "2021-06-11T02:48:11.2851348Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "172.17.0.0/16",
                    "Gateway": "172.17.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {},
        "Options": {
            "com.docker.network.bridge.default_bridge": "true",
            "com.docker.network.bridge.enable_icc": "true",
            "com.docker.network.bridge.enable_ip_masquerade": "true",
            "com.docker.network.bridge.host_binding_ipv4": "0.0.0.0",
            "com.docker.network.bridge.name": "docker0",
            "com.docker.network.driver.mtu": "1500"
        },
        "Labels": {}
    }
]
```

2. Running a container before inspecting

```
PS C:\Users\JervinJosh\Documents\Github\ME8_Containerization_and_Docker\5-volumes> docker run --rm -d --name test jer
vinjosh68/ping:1.0
f11dffb26d486d537da60a782418a439a598465b5e24b0296c70c8332abde572
PS C:\Users\JervinJosh\Documents\Github\ME8_Containerization_and_Docker\5-volumes> docker network inspect bridge
[
    {
        "Name": "bridge",
        "Id": "f426e71b734051434c11ddc409a80aec425dd57c8598c856f193290953e6362f",
        "Created": "2021-06-11T02:48:11.2851348Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "172.17.0.0/16",
                    "Gateway": "172.17.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "f11dffb26d486d537da60a782418a439a598465b5e24b0296c70c8332abde572": {
                "Name": "test",
                "EndpointID": "089061dd155401403ec4d8dbbc8cad11cae61817b62409c90f4628ca94b68cbd",
                "MacAddress": "02:42:ac:11:00:02",
                "IPv4Address": "172.17.0.2/16",
                "IPv6Address": ""
            }
        },
        "Options": {
            "com.docker.network.bridge.default_bridge": "true",
            "com.docker.network.bridge.enable_icc": "true",
            "com.docker.network.bridge.enable_ip_masquerade": "true",
            "com.docker.network.bridge.host_binding_ipv4": "0.0.0.0",
            "com.docker.network.bridge.name": "docker0",
            "com.docker.network.driver.mtu": "1500"
        },
        "Labels": {}
    }
]
```

3. Adding a pinger

```
PS C:\Users\JervinJosh\Documents\Github\ME8_Containerization_and_Docker\5-volumes> docker run --rm -d -e PING_TARGET=172.17.0.2 --name pinger jervinjosh68/ping:1.0
9e45b50193ae5986c9b938c8984ef4ef809701c959ef830c89fefc5616e97d08
PS C:\Users\JervinJosh\Documents\Github\ME8_Containerization_and_Docker\5-volumes> docker ps
CONTAINER ID   IMAGE                  COMMAND                  CREATED         STATUS         PORTS     NAMES
9e45b50193ae   jervinjosh68/ping:1.0   "sh -c 'ping $PING_T…"   7 seconds ago   Up 6 seconds             pinger
f11dffb26d48   jervinjosh68/ping:1.0   "sh -c 'ping $PING_T…"   4 minutes ago   Up 4 minutes             test
PS C:\Users\JervinJosh\Documents\Github\ME8_Containerization_and_Docker\5-volumes> docker logs pinger
PING 172.17.0.2 (172.17.0.2) 56(84) bytes of data.
64 bytes from 172.17.0.2: icmp_seq=1 ttl=64 time=0.925 ms
64 bytes from 172.17.0.2: icmp_seq=2 ttl=64 time=0.044 ms
64 bytes from 172.17.0.2: icmp_seq=3 ttl=64 time=0.045 ms
64 bytes from 172.17.0.2: icmp_seq=4 ttl=64 time=0.079 ms
64 bytes from 172.17.0.2: icmp_seq=5 ttl=64 time=0.071 ms
64 bytes from 172.17.0.2: icmp_seq=6 ttl=64 time=0.089 ms
64 bytes from 172.17.0.2: icmp_seq=7 ttl=64 time=0.118 ms
64 bytes from 172.17.0.2: icmp_seq=8 ttl=64 time=0.063 ms
64 bytes from 172.17.0.2: icmp_seq=9 ttl=64 time=0.044 ms
64 bytes from 172.17.0.2: icmp_seq=10 ttl=64 time=0.048 ms
64 bytes from 172.17.0.2: icmp_seq=11 ttl=64 time=0.047 ms
64 bytes from 172.17.0.2: icmp_seq=12 ttl=64 time=0.081 ms
64 bytes from 172.17.0.2: icmp_seq=13 ttl=64 time=0.046 ms
```

4. Running pinger with test

```
PS C:\Users\JervinJosh\Documents\Github\ME8_Containerization_and_Docker\5-volumes> docker run --rm -d -e PING_TARGET=test --name pinger jervinjosh68/ping:1.0
4b4823f74ddafda6d7190a0eea842679a3916f9fbeb7a773cd0e03486f24c3b3
PS C:\Users\JervinJosh\Documents\Github\ME8_Containerization_and_Docker\5-volumes> docker ps
CONTAINER ID   IMAGE                  COMMAND                  CREATED          STATUS          PORTS     NAMES
f11dffb26d48   jervinjosh68/ping:1.0   "sh -c 'ping $PING_T…"   10 minutes ago   Up 10 minutes             test
```

Managing custom Networks

1. Creating a custom network

```
PS C:\Users\JervinJosh\Documents\Github\ME8_Containerization_and_Docker\5-volumes> docker network create skynet
8bd894fb0e09f8368ef4cb2da19465b56b187ffec25ebe0c477a647a8946d153
PS C:\Users\JervinJosh\Documents\Github\ME8_Containerization_and_Docker\5-volumes> docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
f426e71b7340   bridge    bridge    local
54d52346937f   host      host      local
0a0f7386858c   none      null      local
8bd894fb0e09   skynet    bridge    local
```

```
PS C:\Users\JervinJosh\Documents\Github\ME8_Containerization_and_Docker\5-volumes> docker network inspect skynet
[
    {
        "Name": "skynet",
        "Id": "8bd894fb0e09f8368ef4cb2da19465b56b187ffec25ebe0c477a647a8946d153",
        "Created": "2021-06-11T14:31:49.0497766Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": {},
            "Config": [
                {
                    "Subnet": "172.18.0.0/16",
                    "Gateway": "172.18.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {},
        "Options": {},
        "Labels": {}
    }
]
```

2. Removing a network

```
PS C:\Users\JervinJosh\Documents\Github\ME8_Containerization_and_Docker\5-volumes> docker network rm skynet
skynet
```

Adding Containers to a network

1. Assigning ping to custom network

```
PS C:\Users\JervinJosh\Documents\Github\ME8_Containerization_and_Docker\5-volumes> docker network create skynet
a20284281d1d8c3eaf60521c9217f5aed67c0d1d5102ddb550621b86568003a1
PS C:\Users\JervinJosh\Documents\Github\ME8_Containerization_and_Docker\5-volumes> docker run --rm -d --network skynet --name dummy jervinjosh68/ping:1.0
1a140e74ef43acb0814a41d1d5edf36794ef82b0a346c701929e78c2456a3bf1
```

2. Targeting the ping container

```
PS C:\Users\JervinJosh\Documents\Github\ME8_Containerization_and_Docker\5-volumes> docker run --rm -d --network skynet -e PING_TARGE=dummy --name pinger jervinjosh68/ping:1.0
d1c62ebc0d0c3b27e8840b4edd2ed58c447dfeed82a62a496947d50a256c53e4
PS C:\Users\JervinJosh\Documents\Github\ME8_Containerization_and_Docker\5-volumes> docker logs pinger
PING google.com (172.217.31.238) 56(84) bytes of data.
64 bytes from 172.217.31.238: icmp_seq=1 ttl=37 time=26.0 ms
64 bytes from 172.217.31.238: icmp_seq=2 ttl=37 time=24.9 ms
64 bytes from 172.217.31.238: icmp_seq=3 ttl=37 time=27.1 ms
```

Connecting between containers in a network

1. I had to add the '-e POSTGRES_PASSWORD=password' to make it work

```
PS C:\Users\JervinJosh> docker run --name gadgetdb -e POSTGRES_PASSWORD=password --network skynet -p 5432 -d postgres
ee496a447905c5ff7406e42f6f4b252185a3f6dc094a5d084b0f50feaa1e183c
PS C:\Users\JervinJosh> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED         STATUS         PORTS                     NAMES
ee496a447905   postgres   "docker-entrypoint.s…"   4 seconds ago   Up 3 seconds   0.0.0.0:63798->5432/tcp   gadgetdb
PS C:\Users\JervinJosh> docker run --name widgetdb -e POSTGRES_PASSWORD=password --network skynet -p 5432 -d postgres
5f14a07e3d668dfc1801581bd3cc21c634cf0c292ccea8ad01cfafe606152a6a
PS C:\Users\JervinJosh> docker ps -a
CONTAINER ID   IMAGE      COMMAND                  CREATED         STATUS         PORTS                     NAMES
5f14a07e3d66   postgres   "docker-entrypoint.s…"   6 minutes ago   Up 6 minutes   0.0.0.0:63800->5432/tcp   widgetdb
ee496a447905   postgres   "docker-entrypoint.s…"   7 minutes ago   Up 7 minutes   0.0.0.0:63798->5432/tcp   gadgetdb
```

2. Running widgetdb interactively

```
PS C:\Users\JervinJosh> docker exec -it widgetdb /bin/bas
root@5f14a07e3d66:/#
```

3. Connecting to local database and exiting session

```
root@5f14a07e3d66:/# psql -U postgres
psql (13.3 (Debian 13.3-1.pgdg100+1))
Type "help" for help.

postgres=# \q
root@5f14a07e3d66:/#
```

4. Accessing a different database requires the password to be inputted

```
root@5f14a07e3d66:/# psql -U postgres -h gadgetdb
Password for user postgres:
psql (13.3 (Debian 13.3-1.pgdg100+1))
Type "help" for help.
```

5. Exit and stop

```
postgres=# exit
root@5f14a07e3d66:/# exit
exit
PS C:\Users\JervinJosh> docker stop widgetdb gadgetdb
widgetdb
gadgetdb
```

Binding Ports to Host

1.

```
PS C:\Users\JervinJosh> docker run --rm --name widgetdb -e POSTGRES_PASSWORD=password --network skynet -p 5432:5432 -d p
ostgres
620689fc6faf07644b108a5a8b62886cdb23d990894dbd8e73df88372cda8342
```

2. I had to include the postgresql in the environment variables to make it work

```
PS C:\Users\JervinJosh> psql -U postgres -h localhost
Password for user postgres:
psql (9.5.4, server 13.3 (Debian 13.3-1.pgdg100+1))
WARNING: psql major version 9.5, server major version 13.0.
         Some psql features might not work.
WARNING: Console code page (437) differs from Windows code page (1252)
         8-bit characters might not work correctly. See psql reference
         page "Notes for Windows users" for details.
Type "help" for help.

postgres=#
```